

FUTURE TECH SKILL HUB



Course Contents / Lesson Plan
Course Title: Full Stack Web Developer
Course Duration: 03 Months

Trainer Name	Engr.Abdul Rasheed
Course Title	Full Stack Web Development

<p>Objectives and Expectations</p>	<p>Employable skills and hands-on practice in Website Development</p> <p>The objective of full stack web development is to create a complete web application from start to finish. This includes designing, creating, testing, and deploying the application. Full stack developers are responsible for both the front-end and back-end of the application.</p> <p>The front-end of a web application is what users see and interact with. It includes the HTML, CSS, and JavaScript that make up the user interface. The back-end of a web application is what happens behind the scenes. It includes the server-side code that handles data storage, processing, and security.</p> <p>Full stack developers need to have a strong understanding of both front-end and backend technologies. They also need to be able to work effectively with others on a team.</p> <p>The expectations of full stack web developers vary depending on the company and the project. However, some common expectations include:</p> <ul style="list-style-type: none"> • The ability to work independently and as part of a team • The ability to learn new technologies quickly • The ability to meet deadlines • The ability to work under pressure
---	--

<p>Entry-level of trainees</p>	<p style="text-align: center;">Intermediate / Matric Science</p>
---------------------------------------	---

<p>Course Execution Plan</p>	<p>The total duration of the course: 4 months (16 Weeks) Class hours: 03 hours per day Theory: 20% Practical: 80%</p>
-------------------------------------	--

Course Title: Full Stack Web Development

Learning Outcomes

By the end of this Course, students will be able to

- Understand the fundamentals of web development, including HTML, CSS, and JavaScript.
- Develop front-end web applications using HTML, CSS, and JavaScript.
- Develop back-end web applications using a programming language such as PHP, Python, Flask, Django Java etc.
- Deploy web applications to a production environment.
- Work effectively as part of a team to develop web applications.
- Communicate effectively with both technical and non-technical audiences about web development projects.
- Identify and solve problems in a timely and efficient manner.

Scheduled Weeks	Module Title	Days	Learning Units	Home Assignment
	Advanced Web UI: HTML5 Foundations	Day 1	<ul style="list-style-type: none">• Modern HTML5 Semantic Structure• SEO Meta Tags & Header Optimization• Web Accessibility (ARIA Labels)• HTML Form Validations & Attributes	The SEO Audit: Rebuild a messy non-semantic layout into a professional, accessible multipage structure. Aim for a 100% Lighthouse accessibility score.

Week 1	Advanced Web UI: HTML5 Foundations	Day 2	<ul style="list-style-type: none"> • Embedding Media (Video, Audio, Iframes) • SVG Integration & Optimization • Data Tables & Advanced Lists 	The SEO Audit: Rebuild a messy non-semantic layout into a professional, accessible multipage structure. Aim for a 100% Lighthouse accessibility score.
	Advanced Web UI:	Day 1	<ul style="list-style-type: none"> • CSS Box Model & Specificity • Flexbox Deep Dive (Alignment & Distribution) 	The Grid Gallery: Build a responsive "Pinterest-style" masonry layout using purely CSS

Scheduled Weeks	Module Title	Days	Learning Units	Home Assignment
Week 2	CSS3 Precision	Day 1	<ul style="list-style-type: none"> • CSS Variables for Theme Management 	Grid. Must be fluid from 1 to 6 columns based on screen size.
	Advanced Web UI: CSS3 Precision	Day 2	<ul style="list-style-type: none"> • CSS Grid (Areas & Implicit Grids) • Responsive Design: Fluid Typography & Breakpoints • CSS Animations & Transitions 	The Grid Gallery: Build a responsive "Pinterest-style" masonry layout using purely CSS Grid. Must be fluid from 1 to 6 columns based on screen size.

Week 3	JavaScript Core & Logic	Day 1	<ul style="list-style-type: none"> • Execution Context & Scope Chain • ES6+ (Destructuring, Spread, Rest) • Array Methods (Map, Filter, Reduce) 	The Logic Engine: Develop a multi-category Quiz System with scoring, timers, and dynamic result rendering using Vanilla JS.
	JavaScript Core & Logic	Day 2	<ul style="list-style-type: none"> • Functional Programming Concepts • DOM Manipulation & Event Bubbling • Local Storage Management 	The Logic Engine: Develop a multi-category Quiz System with scoring, timers, and dynamic result rendering using Vanilla JS.
Week 4	JavaScript Async & APIs	Day 1	<ul style="list-style-type: none"> • Asynchronous JS: Event Loop & Callbacks • Promises & Async/Await • Fetch API & Error Handling (Try/Catch) 	The User Directory: Build a realtime searchable user directory fetching live data from an external API with loading skeletons and error states.
	JavaScript Async & APIs	Day 2	<ul style="list-style-type: none"> • Working with JSON Data • Browser APIs (Intersection Observer) 	The User Directory: Build a realtime searchable user directory fetching live data from an external

Scheduled Weeks	Module Title	Days	Learning Units	Home Assignment
-----------------	--------------	------	----------------	-----------------

			<ul style="list-style-type: none"> • Performance: Throttling & Debouncing 	API with loading skeletons and error states.
<div style="border: 1px solid black; padding: 5px; text-align: center; font-weight: bold; font-size: 2em;">Week 5</div>	ReactJS: Fundamentals	Day 1	<ul style="list-style-type: none"> • Intro to React & Virtual DOM • JSX & Functional Components • Props & Component Communication 	The Component Library: Deconstruct a complex dashboard into 15+ reusable atoms and molecules (Inputs, Buttons, Modal skeletons).
	ReactJS: Fundamentals	Day 2	<ul style="list-style-type: none"> • State Management with useState • Conditional Rendering & Lists • Handling Forms in React 	The Component Library: Deconstruct a complex dashboard into 15+ reusable atoms and molecules (Inputs, Buttons, Modal skeletons).
<div style="border: 1px solid black; padding: 5px; text-align: center; font-weight: bold; font-size: 2em;">Week 6</div>	ReactJS: Advanced Hooks	Day 1	<ul style="list-style-type: none"> • useEffect: Lifecycles & Dependencies • useRef: DOM Access & Persistence • Custom Hooks for Data Fetching 	The Persistent To-Do: Build a task manager that utilizes local storage via useEffect to ensure data persists across sessions.
	ReactJS: Advanced Hooks	Day 2	<ul style="list-style-type: none"> • Optimizing with useMemo & useCallback • Advanced Form Handling (Formik/Hook-Form intro) 	The Persistent To-Do: Build a task manager that utilizes local storage via useEffect to ensure data persists across sessions.

Scheduled Weeks	Module Title	Days	Learning Units	Home Assignment
Week 7	ReactJS: Global State	Day 1	<ul style="list-style-type: none"> The Problem of Prop Drilling Context API for Global State useReducer for Complex Logic 	The Multi-Language Site: Create a site with a global toggle that switches all UI text between two languages using Context.
	ReactJS: Global State	Day 2	<ul style="list-style-type: none"> Managing Auth State globally Integrating Global Theme Toggles 	The Multi-Language Site: Create a site with a global toggle that switches all UI text between two languages using Context.
Week 8	ReactJS: Routing SPAs	Day 1	<ul style="list-style-type: none"> React-Router-Dom v6 Setup Nested Routes & Layouts Dynamic Params (/:id) 	The Product Catalog: Build a multi-page gallery with dedicated product detail pages and a 404 page for invalid IDs.
	ReactJS: Routing SPAs &	Day 2	<ul style="list-style-type: none"> Protected/Private Routes Navigation & Redirect Hooks 	The Product Catalog: Build a multi-page gallery with dedicated product detail pages and a 404 page for invalid IDs.

	NodeJS Express & Core	Day 1	<ul style="list-style-type: none"> • Node Event Loop & CommonJS vs ESM • Express Setup & Routing • MVC Architecture Pattern 	The Contact API: Build a server that accepts POST data from a form and saves it to a local JSON file using the 'fs' module.
	NodeJS Express Core	Day 2	<ul style="list-style-type: none"> • Request/Response Objects • Custom Middleware & Error Handling • Restful API 	The Contact API: Build a server that accepts POST data from a form and saves it to a local JSON file using the 'fs' module.

Scheduled Weeks	Module Title	Days	Learning Units	Home Assignment
Week 9			Principles	
	Database: MongoDB & Mongoose	Day 1	<ul style="list-style-type: none"> • NoSQL vs RDBMS • Mongoose Schemas & Models • CRUD Operations with MongoDB 	The Schema Design: Model a database for a Car Rental system and create a script to seed 50 items into the database.

Week 10	Database: MongoDB & Mongoose	Day 2	<ul style="list-style-type: none"> • Data Validation & Sanitization • Querying & Advanced Filtering • Referencing & Embedding Documents 	The Schema Design: Model a database for a Car Rental system and create a script to seed 50 items into the database.
	Backend Security & Auth	Day 1	<ul style="list-style-type: none"> • Password Hashing (Bcrypt) • JWT Strategy (Sign, Verify) • Cookie-based Authentication 	The Gatekeeper: Implement a full Signup/Login flow and create a locked /profile route that only accepts valid JWT tokens.
Week 11	Backend Security & Auth	Day 2	<ul style="list-style-type: none"> • Environment Variables (.env) • Rate Limiting & Security Middleware 	The Gatekeeper: Implement a full Signup/Login flow and create a locked /profile route that only accepts valid JWT tokens.

Scheduled Weeks	Module Title	Days	Learning Units	Home Assignment

Week 12

The MERN Integration

Day 1

- CORS & Proxying in Development
- Connecting React Fetch/Axios to Express
- Handling Auth Tokens in Frontend

The Full-Stack Bridge: Connect your previous Blog UI to your MongoDB backend. Ensure a user can create a post and see it appear in the feed.

The MERN Integration

Day 2

- Frontend Error Handling from API
- Optimistic UI Updates

The Full-Stack Bridge: Connect your previous Blog UI to your MongoDB backend. Ensure a user can create a post and see it appear in the feed.

Week 13

Advanced Features & Media

Day 1

- Multer for Local File Uploads
- Integrating Cloudinary for Image Storage

The Image Uploader: Allow users to upload profile pictures. Save the URL in MongoDB and the file in Cloudinary.

Advanced Features & Media

Day 2

- Backend Search & Indexing
- Pagination Logic

The Image Uploader: Allow users to upload profile pictures. Save the URL in MongoDB and the file in Cloudinary.

	Deployment & CI/CD	Day 1	<ul style="list-style-type: none"> • Building for Production • Deploying Express to 	The Production Launch: Finalize environment variables for production and perform a live end-
Scheduled Weeks	Module Title	Days	Learning Units	Home Assignment
Week 14			Render/Railway <ul style="list-style-type: none"> • Deploying React to Vercel/Netlify 	to-end bug test on the deployed site.
	Deployment & CI/CD	Day 2	<ul style="list-style-type: none"> • Production Performance Audits • Monitoring & Logging 	The Production Launch: Finalize environment variables for production and perform a live endto-end bug test on the deployed site.
	Capstone Project	Build Phase	Comprehensive feature development including DB modeling and API creation.	Final Project: Build and deploy a full-featured Marketplace or Social Platform clone.

Week 15-16

Capstone Project

Polis h Phase

UI Refinement, Bug Fixing, and Final Live Presentation.

Final Project: Build and deploy a full-featured Marketplace or Social Platform clone.